

Package ‘DynareR’

October 12, 2022

Type Package

Title A Seamless Integration of 'R' and 'Dynare'

Version 0.1.3

Maintainer Sagiru Mati <smati@smati.com.ng>

Description

It allows running 'Dynare' program from base R and R Markdown. 'Dynare' is a software platform for handling a wide class of economic models, in particular dynamic stochastic general equilibrium ('DSGE') and overlapping generations ('OLG') models. This package does not only integrate R and Dynare but also serves as a 'Dynare' Knit-Engine for 'knitr' package. The package requires 'Dynare' (<<https://www.dynare.org/>>) and 'Octave' (<<https://www.gnu.org/software/octave/download.html>>). Write all your 'Dynare' commands in R or R Markdown chunk.

Depends R (>= 3.2.3)

Imports knitr (>= 1.20),magrittr, kableExtra

SystemRequirements Dynare, Octave

Suggests rmarkdown

License GPL

URL <https://CRAN.R-project.org/package=DynareR>

BugReports <https://github.com/sagirumati/DynareR/issues>

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-30 06:50:02 UTC

RoxygenNote 7.1.2

Author Sagiru Mati [aut, cre] (<<https://orcid.org/0000-0003-1413-3974>>)

R topics documented:

DynareR-package	2
add_path	3
eng_dynare	4
import_log	5
include_IRF	6
run_dynare	7
run_models	9
set_dynare_version	10
set_octave_path	11
write_dyn	12
write_mod	14
Index	16

DynareR-package	<i>DynareR: A Seamless Integration of 'R' and 'Dynare'</i>
-----------------	--

Description

It allows running 'Dynare' program from base R and R Markdown. 'Dynare' is a software platform for handling a wide class of economic models, in particular dynamic stochastic general equilibrium ('DSGE') and overlapping generations ('OLG') models. This package does not only integrate R and Dynare but also serves as a 'Dynare' Knit-Engine for 'knitr' package. The package requires 'Dynare' (<<https://www.dynare.org/>>) and 'Octave' (<<https://www.gnu.org/software/octave/download.html>>). Write all your 'Dynare' commands in R or R Markdown chunk.

Author(s)

Maintainer: Sagiru Mati <smati@smati.com.ng> ([ORCID](#))

See Also

Useful links:

- <https://CRAN.R-project.org/package=DynareR>
- Report bugs at <https://github.com/sagirumati/DynareR/issues>

Other important functions: [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

add_path	<i>A wrapper for Octave's addpath to add matlab folder.</i>
----------	---

Description

Use this function to add matlab folder. Use this function if Dynare is **NOT** installed in the standard location

Usage

```
add_path(path)
```

Arguments

path	Path to the matlab folder. Default path is /usr/lib/dynare/matlab for Linux, /usr/lib/dynare/matlab for macOS and c:/dynare/x.y/matlab for Windows, where x.y is Dynare version number.
------	---

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

Other important functions: [DynareR](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
library(DynareR)
## Not run:
add_path('/usr/lib/dynare/matlab') # Default for Linux

add_path('c:/dynare/5.1/matlab') # Default for Windows, but 5.1 can change if later version of
# `Dynare` is installed.

add_path('/usr/lib/dynare/matlab') # Default for macOS

## End(Not run)
```

eng_dynare

DynareR: A Seamless Integration of R and Dynare

Description

This package runs on top of knitr to facilitate communication with Dynare. Run Dynare scripts from R Markdown document.

Usage

```
eng_dynare(options)
```

Arguments

options Chunk options, as provided by knitr during chunk execution. Chunk option for this is dynare

Details

The dynare engine can be activated via

```
knitr::knit_engines$set(dynare = DynareR::eng_dynare)
```

This will be set within an R Markdown document's setup chunk.

Value

Set of Dynare (open-source software for DSGE modelling) codes

Author(s)

Sagiru Mati, ORCID: 0000-0003-1413-3974, <https://smati.com.ng>

- Yusuf Maitama Sule (Northwest) University Kano, Nigeria
- SMATI Academy

References

Bob Rudis (2015). Running Go language chunks in R Markdown (Rmd) files. Available at: <https://gist.github.com/hrbrmstr/9a>

Yihui Xie (2019). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.24.

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
knitr::knit_engines$set(dynare = DynareR::eng_dynare)
library(DynareR)
```

import_log	<i>Import dynare log file as a list of R dataframes.</i>
------------	--

Description

Use this function to import dynare log file as a list of R dataframes. The imported list can be accessed via `dynare$modelNmae`.

Usage

```
import_log(path = ".", model = "")
```

Arguments

path	A character string for the path to the dynare log file.
model	Object or a character string representing the name of the Dynare model file (.mod or .dyn extension)

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
## Not run:

library(DynareR)

demo(bkk)

import_log(model="bkk")

# Alternatively, use the path to the log file
```

```
import_log(path="bkk/bkk.log")  
  
# Access the mported list  
  
dynare$bkk  
  
dynare$bkk$moments  
  
knitr::kable(dynare$bkk$decomposition,format='pandoc')  
  
## End(Not run)
```

include_IRF	<i>Embed the graphs of Impulse Response Function (IRF) in R Markdown document</i>
-------------	---

Description

Use this function to include Dynare IRF into the R Markdown document

Usage

```
include_IRF(path = ".", model = "", IRF = "")
```

Arguments

path	A character string for the path to the IRF graph.
model	Object or a character string representing the name of the Dynare model file (.mod or .dyn extension)
IRF	A character string for the name of the Impulse Response Function as defined in the Dynare codes.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

Author(s)

Sagiru Mati, [ORCID: 0000-0003-1413-3974](https://orcid.org/0000-0003-1413-3974)

- Yusuf Maitama Sule (Northwest) University Kano, Nigeria
- SMATI Academy

References

Bob Rudis (2015).Running Go language chunks in R Markdown (Rmd) files. Available at: <https://gist.github.com/hrbrmstr/9a>

Yihui Xie (2019). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.24.

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
## Not run:
library(DynareR)

demo(bkk)

include_IRF(model="bkk", IRF="E_H2")

# The above code fetches the IRF graph from "bkk/bkk/graphs/bkk_IRF_E_H2.pdf"

# Alternatively, the `path` argument can be used as follows

include_IRF(path="bkk/bkk/graphs/bkk_IRF_E_H2.pdf")

## End(Not run)
```

run_dynare

Create and run Dynare mod file

Description

Use this function to create and run Dynare mod file. Use `run_dynare(code="someCode", model="someModel")` if you want the Dynare files to live in the current working directory. Use `run_dynare(run_dynare(code="someCode", model="someModel", path=someDirectory))` if you want the Dynare files to live in the path different from the current working directory (for example, `someDirectory`).

Usage

```
run_dynare(code, model, import_log = FALSE)
```

Arguments

code	Object or a character string representing the set of Dynare codes
model	Object or a character string representing the name of the Dynare model file (.mod or .dyn extension)
import_log	Logical. Whether or not to import dynare log file.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
library(DynareR)
## Not run:
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
*(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
```



```

var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'

# This is "example1" of the `Dynare` example files executed in current working directory

run_dynare(code=DynareCodes,model="example1",import_log=T)

# import_log=T returns the `dynare` log file as a list of dataframes in an environment `dynare`,
# which can be accessed using `dynare$modelName`

dynare$example1

dynare$example1$correlations

dynare$example1$autocorrelation[4,3]

knitr::kable(dynare$example1$moments,format='pandoc')

# This is "example1" of the `Dynare` example files executed in "DynareR/run_dynare/" folder

run_dynare(code=DynareCodes,model="DynareR/run_dynare/example1")

## End(Not run)

```

run_models

*Run multiple **existing** mod or dyn files.*

Description

Use this function to execute multiple **existing** Dynare files. Use `run_models(model='someModel')` if the Dynare files live in the current working directory. Use `run_models(model='someDirectory/someModel')` if the Dynare files live in the path different from the current working directory (for example, `someDirectory`).

Usage

```
run_models(model = "*", import_log = FALSE)
```

Arguments

model	Object or a vector of character strings representing the names of the Dynare model files excluding <code>.mod</code> or <code>.dyn</code> file extension
import_log	Logical. Whether or not to import dynare log file.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
library(DynareR)

## Not run:
demo(agtrend)
demo(bkk)
demo(example1)

# Provide the list of the `Dynare` files in a vector
# Ensure that "agtrend.mod", "bkk.mod" and "example1.mod"
# live in the current working directory

# Copy the dynare files to the current working directory

lapply(c("agtrend","bkk","example1"),\ (x) file.copy(paste0(x,"/","x",".mod"),"."))

run_models(c("agtrend","bkk","example1")) # Run the models in the vector.

run_models() # Run all models in Current Working Directory.

# You can run all models that live in "DynareR/run_dynare/" folder

# Copy the dynare files to the 'DynareR/run_dynare' directory

lapply(c("agtrend","bkk","example1"),\ (x) file.copy(paste0(x,".mod"),"DynareR/run_dynare"))

run_models("DynareR/run_dynare*") # Note the * at the end.

## End(Not run)
```

set_dynare_version *Set Dynare version*

Description

Use this function to set Dynare version

Usage

```
set_dynare_version(dynare_version="")
```

Arguments

dynare_version Character representing Dynare version (for example 6.1, 4.6.1 and so on). This has effect on Windows only.

Value

Character

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
library(DynareR)
## Not run:

# If you want to use the development version of Dynare

set_dynare_version("6-unstable-2022-04-03-0800-700a0e3a") # The development version of Dynare

# If you want to use Dynare version 5.2

set_dynare_version("5.2")

## End(Not run)
```

set_octave_path	<i>Set Octave path</i>
-----------------	------------------------

Description

Use this function to set Octave path

Usage

```
set_octave_path(engine_path="octave")
```

Arguments

engine_path Path to the Octave executable

Value

Character

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [write_dyn\(\)](#), [write_mod\(\)](#)

Examples

```
library(DynareR)
## Not run:
set_octave_path('C:/Program Files/GNU Octave/Octave-6.4.0/mingw64/bin/octave20.exe')

## End(Not run)
```

write_dyn

write a new dyn file.

Description

Use `write_dyn(code="someCode", model="someModel")` if you want the Dynare file to live in the current working directory. Use `write_dyn(code="someCode", model="someDirectory/someModel")` if you want the Dynare file to live in the path different from the current working directory (for example, `someDirectory`).

Usage

```
write_dyn(code, model)
```

Arguments

code	Object or a character string representing the set of Dynare codes
model	Object or a character string representing the name of the Dynare model file (.mod or .dyn extension)

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_mod\(\)](#)

Examples

```

library(DynareR)
## Not run:
dynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho = 0.95;
tau = 0.025;
beta = 0.99;
delta = 0.025;
psi = 0;
theta = 2.95;
phi = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b+1))*c(+1)))
      *(exp(b+1))*alpha*y(+1)+(1-delta)*k);
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'

# This writes "example1" of the `Dynare` example with dyn extension

write_dyn(code=dynareCodes,model="example1")

# This writes "example1" of the `Dynare` example with dyn extension in "DynareR/write_dyn" folder

write_dyn(code=dynareCodes,model="DynareR/write_dyn/example1")

## End(Not run)

```

write_mod	<i>Write a new mod file.</i>
-----------	------------------------------

Description

Use `write_mod(code="someCode", model="someModel")` if you want the Dynare file to live in the current working directory. Use `write_mod(code="someCode", model="someDirectory/someModel")` if you want the Dynare file to live in the path different from the current working directory (for example, `someDirectory`).

Usage

```
write_mod(code, model)
```

Arguments

code	Object or a character string representing the set of Dynare codes
model	Object or a character string representing the name of the Dynare model file (.mod or .dyn extension)

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

Other important functions: [DynareR](#), [add_path\(\)](#), [eng_dynare\(\)](#), [import_log\(\)](#), [include_IRF\(\)](#), [run_dynare\(\)](#), [run_models\(\)](#), [set_dynare_version\(\)](#), [set_octave_path\(\)](#), [write_dyn\(\)](#)

Examples

```
library(DynareR)
## Not run:
dynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b+1))*c(+1)))
    *(exp(b+1)*alpha*y(+1)+(1-delta)*k);
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
```

```
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'

# This writes "example1" of the `Dynare` example with mod extension

write_mod(code=dynareCodes,model="example1")

# This writes "example1" of the `Dynare` example with mod extension in "DynareR/write_mod" folder

write_mod(code=dynareCodes,model="DynareR/write_mod/example1")

## End(Not run)
```

Index

* **documentation**

add_path, [3](#)
DynareR-package, [2](#)
eng_dynare, [4](#)
import_log, [5](#)
include_IRF, [6](#)
run_dynare, [7](#)
run_models, [9](#)
set_dynare_version, [10](#)
set_octave_path, [11](#)
write_dyn, [12](#)
write_mod, [14](#)

* **important functions**

add_path, [3](#)
DynareR-package, [2](#)
eng_dynare, [4](#)
import_log, [5](#)
include_IRF, [6](#)
run_dynare, [7](#)
run_models, [9](#)
set_dynare_version, [10](#)
set_octave_path, [11](#)
write_dyn, [12](#)
write_mod, [14](#)

add_path, [2](#), [3](#), [5](#), [7](#), [8](#), [10–12](#), [14](#)

DynareR, [3](#), [5](#), [7](#), [8](#), [10–12](#), [14](#)
DynareR (DynareR-package), [2](#)
DynareR-package, [2](#)

eng_dynare, [2](#), [3](#), [4](#), [5](#), [7](#), [8](#), [10–12](#), [14](#)

import_log, [2](#), [3](#), [5](#), [5](#), [7](#), [8](#), [10–12](#), [14](#)
include_IRF, [2](#), [3](#), [5](#), [6](#), [8](#), [10–12](#), [14](#)

run_dynare, [2](#), [3](#), [5](#), [7](#), [7](#), [10–12](#), [14](#)
run_models, [2](#), [3](#), [5](#), [7](#), [8](#), [9](#), [11](#), [12](#), [14](#)

set_dynare_version, [2](#), [3](#), [5](#), [7](#), [8](#), [10](#), [10](#), [12](#),
[14](#)

set_octave_path, [2](#), [3](#), [5](#), [7](#), [8](#), [10](#), [11](#), [11](#), [12](#),
[14](#)

write_dyn, [2](#), [3](#), [5](#), [7](#), [8](#), [10–12](#), [12](#), [14](#)
write_mod, [2](#), [3](#), [5](#), [7](#), [8](#), [10–12](#), [14](#)